**TrackNTrace System Analysis and Design**

**Project Overview: Order Tracking and Management System**

A web-based application that enables efficient order placement, tracking, and management. It will facilitate communication between customers and the supply chain to ensure transparency and timely updates on order status.

**Use Cases**

***1.*** ***User Authentication:***

- Users can register, log in, and log out.

- Admins can manage user accounts.

***2. Order Placement:***

- Users can browse products/services.

- Users can add products to a cart.

- Users can place orders from their cart.

***3. Order Tracking:***

- Users can view the status of their order(s).

- Users receive notifications at key milestones.

***4. Order Management (Admin):***

- Admins can update order statuses.

- Admins can manage product inventory.

- Admins can view and manage all orders.

**Functional Requirements**

***1. Authentication System:***

- Secure registration and login process.

- Password encryption and secure session management.

***2. Product Management:***

- CRUD operations for products.

- Inventory tracking and alerts for low stock.

***3. Order Processing:***

- Shopping cart functionality, including add, remove, and update quantities.

- Order placement with order confirmation details.

- Payment processing integration (mockup for this project).

***4. Order Tracking:***

- Real-time updates on order status.

- Email/SMS notifications for order milestones.

***5. Admin Dashboard:***

- Overview of all orders and their statuses.

- Inventory management interface.

- User account management.

**Non-Functional Requirements**

***1. Performance:***

- The system should handle multiple users concurrently without significant degradation in performance.

- Order status updates should be reflected in real-time.

***2. Security:***

- Secure handling of user data and authentication information.

- Compliance with relevant data protection regulations.

***3. Scalability:***

- The system should be scalable, able to handle increased loads by adding resources.

- Database design should support scalability.

***4. Usability:***

- The user interface should be intuitive and easy to use.

- Responsive design for various devices (desktop, tablet, mobile).

***5. Maintainability:***

- The codebase should be well-documented.

- Adopt a modular architecture to facilitate updates and maintenance.

**Test Cases**

***1. User Authentication:***

- Test successful registration, login, and logout.

- Test failed login attempts with incorrect credentials.

***2. Order Placement:***

- Test adding items to the cart and modifying quantities.

- Test placing an order with valid and invalid payment details.

***3. Order Tracking:***

- Test the visibility of real-time order status updates.

- Test the receipt of notifications for order milestones.

***4. Order Management (Admin):***

- Test updating order statuses and verify if changes reflect correctly.

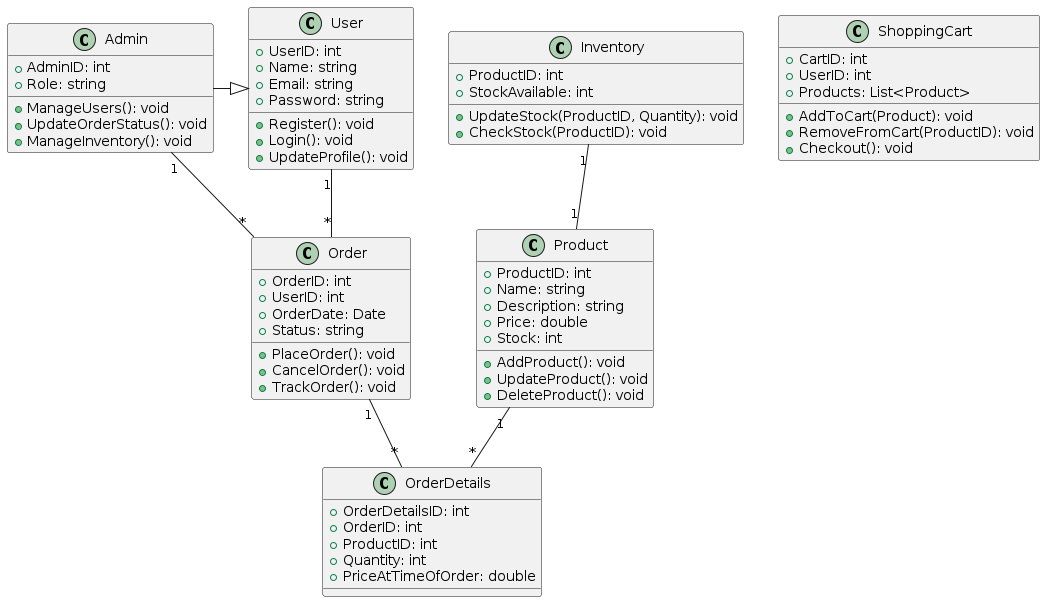
- Test inventory updates and alerts functionality.

***5. Security and Performance:***

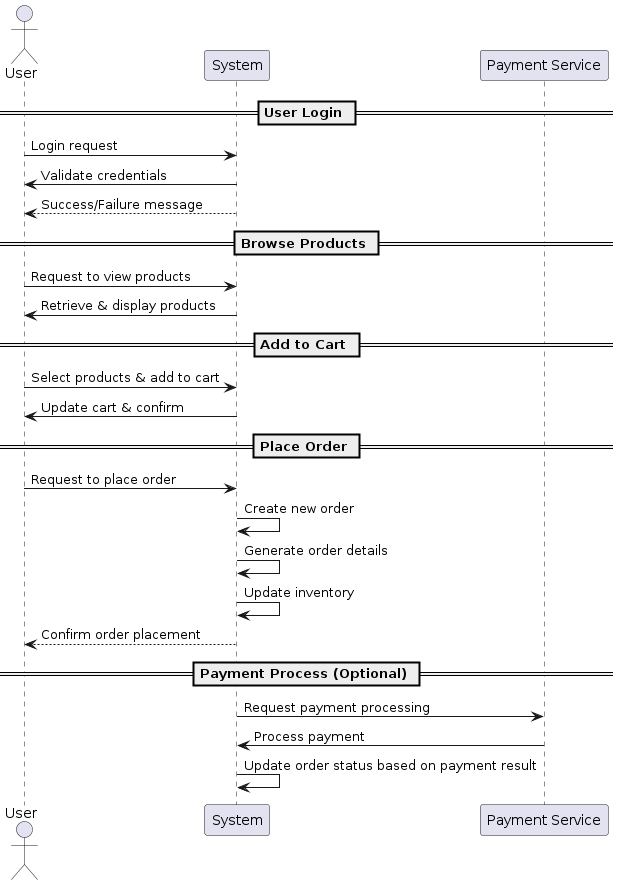
- Conduct security vulnerability assessments.

- Perform load testing to ensure system performance under stress.

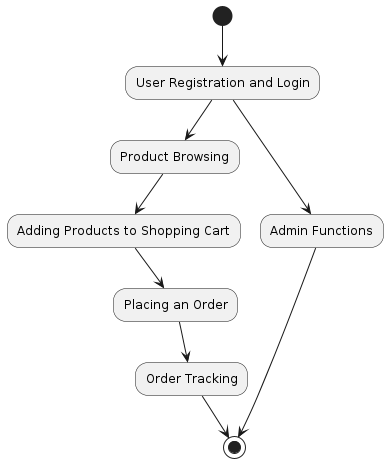
**Class Diagram**



**Sequence Diagram**



**Flow-Chart Diagram**



**Entity Relationship Diagram**

